AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application. No: 09/742,523
Filed: December 20, 2000
Inventor(s):
Steven W. Rogers and Jason S. King
Title: SYSTEM AND METHOD FOR ACCESSING REGISTERS OF A HARDWARE DEVICE IN A GRAPHICAL PROGRAM
Examiner: Ke, Peng
Group/Art Unit: 2174

Atty. Dkt. No: 5150-38200

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed toCommissioner for Patents, Alexandria, VA 22313-1450, on the date indicated below.

Jeffrey C. Hood

4/24/2006
Date
Signature

APPEAL BRIEF

Box: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

Further to the Notice of Appeal filed August 24, 2005 and the Notification of Non-Compliant Appeal Brief mailed on March 24, 2006, Appellant presents this amended Appeal Brief. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The subject application is owned by National Instruments Corporation, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 11500 N. MoPac Expressway, Bldg. B, Austin, Texas 78759-3504.

II. RELATED APPEALS AND INTERFERENCES

No related appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1 – 47 were originally filed in the application. In an amendment filed February 18, 2005, claim 48 was added. Claims 1-48 are pending in the application. All of the pending claims stand rejected and are the subject of this appeal. A copy of the claims incorporating entered amendments, and as on appeal, is included in the Claims Appendix hereto.

IV. STATUS OF AMENDMENTS

All amendments have been entered. No amendments to the claims have been filed subsequent to the rejection in the Office Action of May 20, 2005. The Claims Appendix hereto reflects the current state of the claims.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The independent claims under appeal relate generally to the field of graphical programming. More particularly, the claims relate to configuring a node in a graphical program to access (i.e., read or write) specific hardware registers of a hardware device.

Independent claim 1 recites a method for creating a graphical program which performs hardware register accesses in a hardware device, wherein the method operates in a computer including a display screen and a user input device. The method comprises displaying on the screen a register access node in the graphical program in response to user input. (*See block 80 of Figure 5 and p. 22, lines 6-10*) For example, the user may select the register access node from a palette and drag and drop the register access node onto a window of the graphical program being created. (*See Figure 4 and p. 20, lines 18-22*)

The method further comprises configuring the register access node to access one or more hardware registers of the hardware device. Configuring the register access node to access the one or more hardware registers of the hardware device includes accessing a description of the hardware device for information regarding the one or more hardware registers of the hardware device. (*See block 82 of Figure 5, blocks 202 and 206 of Figure 6, p. 7, lines 14-18, p. 22, lines 11-24, p. 23, lines 6-17, p. 23, line 25 – p. 24, line 2*)

During execution of the graphical program, the register access node is operable to access the one or more hardware registers of the hardware device based on the information regarding the one or more hardware registers of the hardware device. (*See blocks 84 and 86 of Figure 5 and p. 22, line 25 – p. 23, line 2*)

Independent claim 17 recites a similar method for creating a graphical program which performs hardware register accesses in a hardware device, wherein the method operates in a computer including a display and a user input device. The method of claim 17 comprises storing a description of the hardware device. (*See block 202 of Figure 6, p. 7, lines 16-18, and p. 23, lines 6-13*)

The method further comprises displaying on the screen a first node in response to user input, wherein the first node references the description of the hardware device. (*See p. 7, line 26 – p. 8, line 12; See also blocks 204 and 206 of Figure 6 and p. 23, line 18 – p. 24, line 2*) The method further comprises displaying on the screen a register access node in response to user input, wherein the register access node is operable to access the hardware device during execution of the graphical program. (*See p. 7, line 26 – p. 8, line 12; See also block 212 of Figure 6 and p. 24, lines 17-18*) The method further comprises connecting the first node to the register access node in response to user input, wherein the first node is operable to provide the description of the hardware device to the register access node. (*See p. 7, line 26 – p. 8, line 12; See also blocks 210 and 214 of Figure 6 and p. 24, lines 6-25; See also p. 25, lines 23-28*)

Thus, the register access node receives the description of the hardware device from the first node that is connected to the register access node. The register access node is operable to access hardware registers of the hardware device during execution of the graphical program, based on the description of the hardware device. (*See blocks 84 and 86 of Figure 5 and p. 22, line 25 – p. 23, line 2*)

Independent claim 36 recites a similar method as claim 17 for creating a graphical program which performs hardware register accesses in a hardware device, wherein the method operates in a computer including a display and a user input device. The method of claim 36 comprises storing a description of the hardware device. (*See block 202 of Figure 6, p. 7, lines 16-18, and p. 23, lines 6-13*)

The method further comprises displaying on the screen a register access node in the graphical program in response to user input, wherein the register access node is operable to access the hardware device. (*See block 80 of Figure 5 and p. 22, lines 6-10; See also block 212 of Figure 6 and p. 24, lines 17-18*)

The method further comprises connecting an input of the register access node to receive the description of the hardware device in response to user input. (*See p. 7, line 26 – p. 8, line 12; See also blocks 210 and 214 of Figure 6 and p. 24, lines 6-25*)

The method further comprises configuring the register access node to access selected hardware registers described in the description of the hardware device in response to user input. Configuring the register access node includes accessing the description of the hardware device for information regarding the selected hardware registers of the hardware device. (*See block 216 of Figure 6; p. 8, lines 13-14; p. 24, lines 26-29*)

The register access node is operable to access the selected hardware registers of the hardware device during execution of the graphical program, based on the information regarding the selected hardware registers of the hardware device. (*See blocks 84 and 86 of Figure 5 and p. 22, line 25 – p. 23, line 2*)

Independent claim 37 is a memory medium claim analogous to the method claim 1 and recites similar limitations as claim 1. (*See specification and drawing references noted above for claim 1*)

Independent claim 43 is a system claim analogous to the method claim 1 and recites similar limitations as claim 1. (*See specification and drawing references noted above for claim 1*)

Independent claim 48 recites a similar method as claim 1 for creating a graphical program which performs hardware register accesses in a hardware device, wherein the method operates in a computer including a display screen and a user input device. A register access node is displayed in the graphical program in response to user input. (*See block 80 of Figure 5 and p. 22, lines 6-10*) The register access node is configured to access one or more hardware registers of the hardware device. (*See block 82 of Figure 5 and p. 22, lines 11-13*) During execution of the graphical program, the register access node is operable to access the one or more hardware registers of the hardware device. (*See blocks 84 and 86 of Figure 5 and p. 22, line 25 – p. 23, line 2*)

VI. <u>GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL</u>

Claims 1, 3, 5-11, 13-16, 36, 37, 39-43, 45-47, and 48 stand rejected under 35 U.S.C. 103(a) as being unpatentable over McKaskle (U.S. Patent No. 5,481,741) in view of Lee et al. (U.S. Patent No. 5,214,753).

Claims 2, 38, and 44 stand rejected under 35 U.S.C. 103(a) as being unpatentable over McKaskle (U.S. Patent No. 5,481,741) in view of Lee et al. (U.S. Patent No. 5,214,753), in further view of Sojoodi et al. (U.S. Patent No. 5,847,953).

Claims 4, 12, 17, 18, 23-25, 29, and 30-35 stand rejected under 35 U.S.C. 103(a) as being unpatentable over McKaskle (U.S. Patent No. 5,481,741) in view of Lee et al. (U.S. Patent No. 5,214,753), in further view of Yamamoto et al. (U.S. Patent No. 5,847,953).

Claims 19-22 and 26-28 stand rejected under 35 U.S.C. 103(a) as being unpatentable over McKaskle (U.S. Patent No. 5,481,741) in view of Lee et al. (U.S. Patent No. 5,214,753), in further view of Yamamoto et al. (U.S. Patent No. 5,847,953), in further view of McIntyre et al. (U.S. Patent No. 6,229,538).

VII. ARGUMENT

Section 103(a) Rejections Based on McKaskle and Lee

Claims 1, 3, 5-11, 13-16, 36, 37, 39-43, and 45-48 stand rejected under 35 U.S.C. 103(a) as being unpatentable over McKaskle et al. (U.S. Patent No. 5,481,741, hereinafter "McKaskle") in view of Lee et al. (U.S. Patent No. 5,214,753, hereinafter "Lee"). Appellant respectfully traverses this rejection.

Claim 1

As per claim 1, the claim recites as follows:

> 1. (Previously Presented) A method for creating a graphical program which performs hardware register accesses in a hardware device, wherein the method operates in a computer including a display screen and a user input device, the method comprising:
>
> displaying on the screen a register access node in the graphical program in response to user input; and
>
> configuring the register access node to access one or more hardware registers of the hardware device, wherein said configuring includes accessing a description of the hardware device for information regarding the one or more hardware registers of the hardware device;
>
> wherein, during execution of the graphical program, the register access node is operable to access the one or more hardware registers of the hardware device based on the information.

The Examiner asserts that the element of "configuring the register access node to access one or more hardware registers of the hardware device" is taught by McKaskle in Fig. 99 and Col. 49, lines 49-56. Appellant disagrees.

Fig. 99 illustrates a While loop structure. A block diagram can be placed inside the border of the While loop, and the While loop executes the block diagram inside it for one or more iterations until a Boolean value passed to the conditional terminal of the While loop is False. (See col. 48, lines 31-42.) As explained in the cited passage at Col. 49, users can create local variables referred to as "shift registers" to transfer values from one iteration of a While loop (or For loop) to the next by right-clicking on the left or right border of the loop and selecting "Add Shift Register" from the pop-up menu. The shift register contains a pair of terminals displayed directly opposite each other on the left vertical side of the loop border and the right vertical side of the loop border. Data can be passed from the block diagram inside the loop border to the shift register terminal on the right side of the loop border, e.g., by connecting a wire from an output terminal of a node in the block diagram to the right shift register terminal. The data passed to the right shift register terminal is then "shifted" to the left shift register terminal so that it is available for use in the next iteration of the loop. For example, the left shift register terminal can be wired to an input terminal of a node in the block diagram inside the border of the loop to cause the node to receive the data that was passed to the right shift register terminal in the previous iteration of the loop.

Thus, as McKaskle states, "shift registers (available for While Loops and For Loops) are local variables that transfer values from one iteration to the next" (Col. 49, lines 34-36). McKaskle also teaches that, "A shift register can hold any data type – numeric, Boolean, string, array, and so on. The shift register automatically adapts to the data type of the first object that is wired to the shift register" (Col, 49, lines 44-47). Thus, a shift register is essentially a software variable, which is simply not the same as a hardware register of a hardware device. A shift register is a software programming construct that does not even exist until the user right-clicks on the loop border to select the "Add Shift Register" menu item from the pop-up menu. In contrast, a hardware register of a hardware device is a tangible physical component of the hardware device and exists independently of any action performed by a user. Furthermore, a hardware register of a hardware device typically holds a fixed number of bits and does not "adapt" to different data types, as McKaskle teaches that a shift register does.

Claim 1 further recites that configuring the register access node to access the one or more hardware registers of the hardware device includes "accessing a description of the hardware device for information regarding the one or more hardware registers of the hardware device". This element of the claim further clarifies that the one or more hardware registers referred to in claim 1 are tangible physical components of the hardware device and are not the same as software variables. Furthermore, Appellant disagrees with the Examiner's assertion that this element of the claim is taught by Lee at Col. 13, lines 1-27. Lee relates generally to a circuit for a video system (Abstract). In particular, the cited portion of Lee relates generally to the use of a register 260 in a video circuit, where the register stores a horizontal position value used in an equation. There is nothing at all in the cited portion regarding accessing a description of a hardware device for information regarding one or more hardware registers of the hardware device.

Claim 1 further recites, "wherein, during execution of the graphical program, the register access node is operable to access the one or more hardware registers of the hardware device based on the information." The Examiner asserts that this element of the claim is taught by McKaskle at Col. 26, lines 2-27. The cited portion relates to a graphical programming construct called a "sequence structure," an example of which is illustrated in FIG. 12A. A user can place multiple block diagrams inside the border of the sequence structure, and the block diagrams are executed in sequence during execution of the graphical program (Col. 25, lines 45-52). The passage cited by the Examiner relates to FIG. 13, which illustrates conceptually how a sequence structure operates to cause the multiple block diagrams to execute in sequence. McKaskle teaches that "input registers" are provided to collect input data for the sequence structure from other nodes in the graphical program. When all data inputs are present, the sequence structure begins executing the first diagram in the sequence (Col. 25, lines 53-54 and 62-67; Col. 26, lines 10-13).

Appellant respectfully submits that the sequence structure is a graphical software programming construct (see FIG. 12A; Col. 7, lines 10-11 and 20-22), and the "input registers" referred to in this context are essentially software variables that receive data passed to the sequence structure from other nodes in the graphical program. Thus, the cited portion of McKaskle does not teach accessing one or more hardware registers of a hardware device during execution of a graphical program.

Furthermore, Appellant notes that the input registers referred to in the context of a sequence structure are not the same as the shift registers of a While loop, and thus, the Examiner has not been consistent in which elements of McKaskle are being asserted as equivalent to the "one or more hardware registers of a hardware device" recited in claim 1.

Thus, for at least the reasons provided above, Appellant submits that McKaskle and Lee, taken either singly or in combination, do not teach numerous elements of claim 1. Appellant further submits that the Examiner has not established a case of *prima facie* obviousness for several reasons. For example, Lee is not analogous art with Appellant's invention. Lee relates generally to circuitry for a video system, whereas Appellant's invention relates generally to the field of graphical software programming, and more particularly to a system and method for accessing registers of hardware devices from a graphical program. Appellant respectfully submits that a person wanting to provide a system and method for accessing registers of a hardware device from a graphical program would not look to the art of video circuitry.

Appellant also submits that the Examiner has not given a reasonable motivation for combining McKaskle and Lee. The Examiner states that, "It would have been obvious to an artisan at the time of the invention to include Lee's teaching with method of McKaskle in order to provide the user with the ability to select position data of register that can be linked to the bus." However, the passage cited in Lee relates to a circuit in a video system for calculating intersection points between scan lines (Col. 12, lines 1-9 and 65-66). Appellant respectfully submits that it makes little or no sense to suggest that one would want to provide a user with the ability to select position data stored in a register in a video circuit that calculates intersection points between scan lines, and further submits that there is not even a user involved in the operation of Lee's video circuit. Appellant also submits that there is no teaching actually found in the prior art that suggests to make this combination.

Thus, for at least the reasons given above, Appellant respectfully submits that claim 1 is patentable over McKaskle and Lee. Furthermore, if an independent claim is non-obvious under 35 U.S.C. 103, then any claim depending therefrom is also non-obvious. Thus Appellant respectfully submits that the claims dependent on claim 1 are also non-obvious, and thus patentable over the cited references. Appellant further submits that numerous ones of the dependent claims recite further distinctions over the cited art, as described below.

Claim 3

Claim 3 is separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitations recited in this claim. Claim 3 adds to claim 1 the elements of: "storing the description of the hardware device in the computer;

wherein the register access node uses the description of the hardware device to access hardware registers of the hardware device during execution of the graphical program."

The Examiner asserts that the limitations of claim 3 are taught by McKaskle at Col. 31, lines 34-67. However, Appellant notes that in the rejection of claim 1, the Examiner admits that, "McKaskle fails to teach said configuring includes accessing a description of the hardware device for information regarding the one or more registers of the hardware devices." Since McKaskle does not teach accessing a description of the hardware device for information regarding the one or more hardware registers of the hardware device, Appellant submits that McKaskle also does not teach a register access node using the description of the hardware device to access hardware registers of the hardware device.

Furthermore, the portion of McKaskle cited in the rejection of claim 3 relates very generally to the concepts of graphical programming and data flow in a graphical program. The cited portion teaches nothing whatsoever about a description of a hardware device, where the description includes information regarding hardware registers of the hardware device. The cited portion also teaches nothing at all about a node in a graphical program accessing a hardware register of a hardware device during execution of a graphical program. More particularly, the cited portion teaches nothing about a node in a graphical program using a description of a hardware device to access hardware registers of the hardware device during execution of the graphical program.

Thus, Appellant respectfully submits that McKaskle does not teach the limitations recited in claim 3, and thus, claim 3 is patentable over McKaskle and Lee.

Claim 5

Claim 5 is separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitations recited in this claim. Claim 5 adds to claim 3 the elements of:

> wherein the description includes mnemonic names of hardware registers;
> wherein said configuring the register access node includes:
> displaying a list of the mnemonic names of hardware registers on the display; and
> receiving user input selecting one or more of the mnemonic names of hardware registers for access.

The Examiner asserts that the limitations of claim 5 are taught by McKaskle in FIG. 90 and refers to the illustrated graphical program nodes labeled "DBL". However, FIG. 90 illustrates an example of a Help window (Col. 48, lines 22-48) for a node that performs a math addition function. The "DBL" label means that the respective inputs and outputs have a data type of "Double". FIG. 90 has nothing whatsoever to do with displaying a list of mnemonic names of hardware registers or receiving user input selecting one or more of the mnemonic names. McKaskle simply does not teach the limitations recited in claim 5, either in FIG. 90 or elsewhere. Appellant thus submits that claim 5 is patentable over McKaskle and Lee.

Claim 6

Claim 6 is separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitations recited in this claim. Claim 6 adds to claim 5 the elements of:

> wherein the description further includes mnemonic names of fields in the hardware registers;
>
> wherein said configuring the register access node includes:
>
> displaying a list of the mnemonic names of fields in the hardware registers on the display; and
>
> receiving user input selecting one or more of the mnemonic names of fields in the hardware registers for access.

The Examiner asserts that the limitations of claim 6 are taught by McKaskle in FIG. 89 and refers to the illustrated menu items. However, McKaskle clearly teaches that FIG, 89 illustrates a "Functions" menu that includes a "VI..." menu item that can be used to select subVI's to be used in the block diagram of another VI. FIG. 89 has nothing whatsoever to do with displaying a list of mnemonic names of fields in hardware registers or receiving user input selecting one or more of the mnemonic names of fields in the hardware registers. McKaskle simply does not teach the limitations recited in claim 6, either in FIG. 89 or elsewhere. Appellant thus submits that claim 6 is patentable over McKaskle and Lee.

Claims 7 and 41

Claims 7 and 41 are separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitations recited in these claims. For example, claim 7 adds to claim 1 the elements of:

> wherein said configuring the register access node comprises:
> displaying an icon on the graphical program which references register access node configuration information; and
> connecting the icon to the register access node.

The Examiner asserts that the limitations of claim 7 are taught by McKaskle in FIG. 19g and refers to the graphical program objects labeled "DBL" and "Wave". However, FIG. 19g is simply part of a series of figures illustrating construction of an exemplary block diagram and does not teach the specific limitations recited in claims 7 and 41. McKaskle does not teach displaying an icon on the graphical program which references register access node configuration information and connecting the icon to the register access node. Appellant thus submits that claims 7 and 41 are patentable over McKaskle and Lee.

Claim 8

Claim 8 is separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitation recited in this claim. Claim 8 adds to claim 7 the limitation of, "wherein the icon is a hardware open node." The Examiner asserts that the limitation of claim 8 is taught by McKaskle in FIG. 19g and refers to the graphical program objects labeled "DBL" and "Wave". However, McKaskle teaches that the object labeled "DBL" is a terminal associated with a control knob placed in the front panel (user interface) (Col. 29, lines 36-38). McKaskle also teaches that the object labeled "Wave" references another built-in VI, but does not describe the function of the VI (Col. 30, lines 4-5). McKaskle does not teach that any of the objects in FIG. 19g is a hardware open node such as described in the present application. Appellant thus submits that claim 8 is patentable over McKaskle and Lee.

Claim 9

Claim 9 is separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitations recited in this claim. Claim 9 adds to claim 1 the elements of:

> wherein said register access node comprises one or more input terminals, the method further comprising:
>
> configuring the one or more input terminals to write to a hardware register of the hardware device.

The Examiner asserts that the limitations of claim 9 are taught by McKaskle in FIG. 80; Col. 64, lines 15-21; and Col. 46, lines 16-34. However, neither the cited passages nor FIG. 80 teach anything whatsoever about writing to a hardware register of a hardware device, and in particular, do not teach configuring one or more input terminals of a register access node to write to a hardware register of a hardware device. McKaskle simply does not teach the limitations recited in claim 9. Appellant thus submits that claim 9 is patentable over McKaskle and Lee.

Claim 10

Claim 10 is separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitations recited in this claim. Claim 10 adds to claim 1 the elements of:

> wherein said register access node comprises one or more output terminals, the method further comprising:
>
> configuring the one or more output terminals to read a hardware register of the hardware device.

The Examiner asserts that the limitations of claim 10 are taught by McKaskle in FIGS. 111 and 112, and Col. 51, lines 25-49. However, the cited portions of McKaskle relate to a waveform chart, which is a GUI indicator to display one or more plots. The cited portions do not teach anything whatsoever about reading a hardware register of a hardware device, and in particular, do not teach configuring one or more output terminals of a register access node to read a hardware register of a hardware device. McKaskle simply does not teach the limitations recited in claim 10. Appellant thus submits that claim 10 is patentable over McKaskle and Lee.

Claim 11

Claim 11 is separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitations recited in this claim. Claim 11 adds to claim 1 the elements of:

> displaying on the screen a first node in response to user input, wherein the first node references the hardware device; and
>
> connecting the first node to the register access node, wherein said connecting provides the register access node with information regarding the hardware device.

The Examiner asserts that the limitations of claim 11 are taught by McKaskle in FIG. 80; Col. 64, lines 15-21; and Col. 46, lines 16-34. However, the cited portions of McKaskle do not teach the specific limitations recited in claim 11. McKaskle does not teach connecting a first node to a register access node, wherein the first node references a hardware device, and wherein said connecting provides the register access node with information regarding the hardware device. Appellant thus submits that claim 11 is patentable over McKaskle and Lee.

Claims 14, 15, 42, and 47

Claims 14, 15, 42, and 47 are separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitations recited in these claims. Claim 14 adds to claim 1 the limitation of, "constructing execution instructions in response to the graphical program, wherein the execution instructions are executable to access hardware registers of the hardware device." Claim 15 adds to claim 14 the further limitation of, "executing said execution instructions, wherein the register access node accesses hardware registers of the hardware device during said executing." The Examiner asserts that the limitations of both claim 14 and claim 15 are taught by McKaskle at Col. 31, lines 34-55. The cited portion of McKaskle relates generally to the concept of graphical programming, where execution instructions are constructed in response to a graphical block diagram created by a user. However, the cited portion teaches nothing whatsoever about constructing execution instructions executable to access hardware registers of a hardware device or executing the execution instructions, wherein a register access node accesses hardware registers of the hardware device during the executing. McKaskle simply does not teach the specific limitations recited in claims 14 and 15.

Appellant thus submits that claims 14 and 15 are patentable over McKaskle and Lee. Claims 42 and 47 recite similar limitations as claim 14, and thus are also patentable over McKaskle and Lee.

Claim 16

Claim 16 is separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitations recited in this claim. Claim 16 adds to claim 1 the limitation of, "wherein the graphical program is operable to access hardware registers of the hardware device for performing instrumentation functions on an instrument." The Examiner asserts that this limitation of claim 16 is taught by McKaskle at Col. 33, lines 49-61. However, the cited portion of McKaskle relates to setting a breakpoint so that a VI enters a suspended state when it is about to execute. This portion of McKaskle teaches nothing whatsoever about a graphical program accessing hardware registers of a hardware device, and more particularly, teaches nothing about accessing hardware registers of a hardware device for performing instrumentation functions on an instrument. McKaskle simply does not teach the specific limitation recited in claim 16.

Claim 36

Independent claim 36 includes similar limitations as recited in claim 1 and various claims dependent on claim 1, the rejections of which are addressed above. Appellant thus submits that the arguments set forth above with respect to claim 1 and its dependent claims apply with equal force to claim 36 and that claim 36 is patentable over McKaskle and Lee for similar reasons.

Claims 37, 43, and 48

Independent claims 37 and 43 are memory medium and system claims, respectively, analogous to the method claim 1 and include similar limitations. Independent claim 48 is a method claim reciting similar limitations as claim 1. Appellant thus submits that the arguments set forth above with respect to claim 1 apply with equal force to claims 37, 43, and 48, and that these claims are also patentable over McKaskle and Lee for similar reasons.

Claims 39 and 45

Claims 39 and 45 are separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitations recited in these claims. For example, claim 39 recites:

> 39. (Previously Presented) The memory medium of claim 37, further comprising program instructions executable by the processor to:
>
> store the description of the hardware device in the computer;
>
> wherein the register access node uses the description of the hardware device to access hardware registers of the hardware device during execution of the graphical program.

Appellant notes that claims 39 and 45 recite similar limitations as claim 3, the rejection of which was traversed above. However, in the rejection of claims 39 and 45, the Examiner cites a different portion of McKaskle, asserting that the limitations are taught at Col. 14, lines 63-68. However, this portion of McKaskle relates to FIG. 8A, which illustrates various software components of a virtual instrument (graphical program), such as a block diagram, front panel (user interface), etc. Appellant submits that none of the components illustrated in FIG. 8A is a description of a hardware device, wherein the description is useable by a node to access hardware registers of the hardware device during execution of a graphical program, and also submits that McKaskle contains no teaching whatsoever regarding the limitations recited in claims 39 and 45.

Claims 40 and 46

Claims 40 and 46 are separately patentable over McKaskle and Lee because the references do not teach or suggest the further limitations recited in these claims. For example, claim 40 adds to claim 39 the limitation of, "wherein said configuring the register access node to access one or more hardware registers of the hardware device comprises configuring the register access node to access selected hardware registers described in the description of the hardware device." Appellant notes that claim 4 recites a similar limitation as claims 40 and 46. The Examiner admits in the rejection of claim 4 that, "However, they [McKaskle and Lee] fail to teach the method wherein said configuring the register access node to access one or more registers of the hardware device comprises configuring the register access node to access selected registers described in the description of the hardware device." Appellant thus submits that the Examiner has explicitly admitted that McKaskle and Lee fail to teach the limitation recited in claims 40 and 46, but the Examiner has still rejected claims 40 and 46 based on McKaskle and Lee. Appellant further submits that the portions of McKaskle cited in the rejection of claims 40 and 46 (FIG. 99 and Col. 49, lines 49-56) do not teach the limitation recited in claims 40 and 46, and thus, claims 40 and 46 are patentable over McKaskle and Lee.

Section 103(a) Rejections Based on McKaskle, Lee, and Sojoodi

Claims 2, 38, and 44 stand rejected under 35 U.S.C. 103(a) as being unpatentable over McKaskle in view of Lee, further in view of Sojoodi et al. (U.S. Patent No. 5,847,953, hereinafter "Sojoodi"). Appellant respectfully traverses this rejection.

Claim 2 recites "The method of claim 1, wherein said configuring the register access node includes: displaying a list of hardware registers; and receiving user input to select one or more of the hardware registers from the list of hardware registers." The Examiner asserts that these limitations are taught by Sojoodi in FIG. 7 and Col. 17, lines 17-46. Appellant disagrees.

Sojoodi relates generally to performing virtual instrumentation functions in a device resource independent manner in a graphical data flow program (Col. 1, lines 64-67). FIG. 7 illustrates a function node palette that provides access to various VISA (Virtual Instrumentation Software Architecture) function nodes and illustrates a help window for a VISA Write function node for writing data to a device. FIG. 7 does not display a list of hardware registers as recited in claim 2. Furthermore, no user input to select one or more hardware registers is received to any of the windows shown in FIG. 7. Appellant also notes that since Sojoodi is targeted toward solving the problem of performing virtual instrumentation functions in a <u>device resource independent manner</u>, there would be no reason to configure a function node to perform an instrumentation function by displaying and selecting specific hardware registers for the function to operate on, since this would be a device-specific manner of configuring the function node, not a device-independent manner.

Appellant thus submits that the cited references do not teach the limitations recited in claim 2, and thus, claim 2 is patentable over McKaskle, Lee, and Sojoodi. Inasmuch as claims 38 and 44 recite similar limitations as claim 2, these claims are also patentable over McKaskle, Lee, and Sojoodi.

Section 103(a) Rejections Based on McKaskle, Lee, and Yamamoto

Claims 4, 12, 17, 18, 23-25, 29, and 30-35 stand rejected under 35 U.S.C. 103(a) as being unpatentable over McKaskle in view of Lee, further in view of Yamamoto et al. (U.S. Patent No. 6,553,431, hereinafter "Yamamoto"). Appellant respectfully traverses this rejection.

Claim 4

Claim 4 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 4 adds to claim 3 the limitation of, "wherein said configuring the register access node to access one or more hardware registers of the hardware device comprises configuring the register access node to access selected hardware registers described in the description of the hardware device."

The Examiner asserts that this limitation is taught by Yamamoto at Col. 10, lines 12-25. The cited portion of Yamamoto relates to the steps S15, S16, and S17 in the flowchart illustrated in FIG. 8. The flowchart of FIG. 8 shows an input/output device selection procedure in which the input and output devices for a copying operation are virtually selected. The steps S15, S16, and S17 relate to acquiring a device profile of an output device. The Examiner is apparently asserting equivalence between this device profile and the hardware device description of claim 4. However, Yamamoto does not teach that the device profile describes hardware registers of a hardware device, as required by claim 4. For example, FIG. 7 of Yamamoto illustrates a device profile that contains information about a device, such as a device type, device ID, network address, etc., but does not contain information describing hardware registers of the device.

Furthermore, Yamamoto contains no teaching whatsoever about a node in a graphical program, and more particularly, contains no teaching about configuring a register access node to access selected hardware registers described in a description of a hardware device.

Appellant thus submits that the cited references do not teach the limitations recited in claim 4, and thus, claim 4 is patentable over McKaskle, Lee, and Yamamoto.

Claims 12 and 31

Claims 12 and 31 are separately patentable over the cited references because the references do not teach or suggest the further limitations recited in these claims. For example, claim 12 adds to claim 11 the limitation of, "wherein the first node is a hardware refnum node which references the description of the hardware device."

The Examiner asserts that this limitation is taught by Yamamoto in FIG. 9A and Col. 10, lines 37-68. However, Yamamoto is not related to the field of graphical programming and does not disclose the concept of a node in a graphical program, either in the portions cited by the Examiner or elsewhere. FIG. 9A illustrates a relationship among devices, where the devices are represented by icons, but these icons are not nodes in an executable graphical program. Thus, Yamamoto certainly does not teach a hardware refnum node in a graphical program, such as described in the present application.

Appellant thus submits that the cited references do not teach the limitations recited in claim 12, and thus, claim 12 is patentable over McKaskle, Lee, and Yamamoto. Inasmuch as claim 31 recites a similar limitation as claim 12, Appellent submits that claim 31 is also patentable over McKaskle, Lee, and Yamamoto.

Claim 17

Claim 17 is separately patentable over the cited references because the references do not teach or suggest the limitations recited in this claim, for reasons similar to those discussed above with reference to various claims, e.g., claims 1, 3, and 11.

Claim 18

Claim 18 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 18 adds to claim 17 the limitation of, "configuring the register access node to access selected hardware registers described in the description of the hardware device, wherein said configuring includes accessing the description of the hardware device for information regarding the selected hardware registers."

The Examiner asserts that this limitation is taught by Yamamoto in FIG. 27A, referring to various icons. However, as discussed above with reference to claim 12, these icons are not nodes in a graphical program, but instead, are simply icons representing various devices. Thus, FIG. 27A does not teach configuring a register access node, which is a node in a graphical program. Furthermore, FIG. 27A teaches nothing whatsoever about accessing a description of a hardware device for information regarding the hardware registers.

Appellant thus submits that the cited references do not teach the limitations recited in claim 18, and thus, claim 18 is patentable over McKaskle, Lee, and Yamamoto.

Claim 23

Claim 23 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 23 adds to claim 17 the limitations of, "displaying on the screen a list of descriptions of available hardware devices" and "selecting the description from the list of descriptions."

The Examiner states that claim 23 is of the same scope as claim 18. Appellant respectfully disagrees and submits that claim 23 recites different limitations than claim 18. Appellant also submits that the cited references do not teach displaying a list of descriptions of available hardware devices and selecting a description of a hardware device from the displayed list, wherein a node accesses hardware registers of a hardware device during execution of a graphical program based on the selected description, as recited in claims 17 and 23. Appellant thus submits that claim 23 is patentable over McKaskle, Lee, and Yamamoto.

Claim 24

Claim 24 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 24 adds to claim 17 the limitations of, "wherein said register access node comprises one or more input terminals, wherein, for each input terminal, the method further comprises: configuring each input terminal to write to a hardware register of the hardware device."

The Examiner asserts that these limitations are taught by Yamamoto in FIG. 27A, referring to various icons. However, as discussed above with reference to claims 12 and 18, these icons are not nodes in a graphical program, but instead, are simply icons representing various devices. Thus, FIG. 27A does not teach configuring input terminals of a register access node, which is a node in a graphical program. Furthermore, FIG. 27A teaches nothing whatsoever about writing to a hardware register of a hardware device. Yamamoto teaches that FIGS. 27A, 27B and 27C illustrate an example of a procedure to form virtual input and output devices by a graphical user interface (Col. 5, lines 23-25).

Appellant thus submits that the cited references do not teach the limitations recited in claim 24, and thus, claim 24 is patentable over McKaskle, Lee, and Yamamoto.

Claim 25

Claim 25 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 25 adds to claim 17 the limitations of, "wherein said register access node comprises one or more output terminals, wherein, for each output terminal, the method further comprises: configuring said each output terminal to read a hardware register of the hardware device."

The Examiner asserts that these limitations are taught by Yamamoto at Col. 12, lines 35-49. However, the cited portion of Yamamoto teaches nothing whatsoever about configuring an output terminal of a node in a graphical program to read a hardware register of a hardware device.

Appellant thus submits that the cited references do not teach the limitations recited in claim 25, and thus, claim 25 is patentable over McKaskle, Lee, and Yamamoto.

Claim 29

Claim 29 is separately patentable over the cited references because the references do not teach or suggest the further limitation recited in this claim. Claim 29 adds to claim 17 the limitation of, "wherein the first node is a hardware open node." The Examiner asserts that the limitation of claim 29 is taught by McKaskle in FIG. 100B and refers to the object labeled "7". However, this object is simply a numeric value. McKaskle contains no teaching whatsoever about this object being a hardware open node such as described in the present application. Furthermore, the object labeled "7" is not connected to a register access node and does not provide a description of a hardware device to a register access node, as required by claims 17 and 29.

Appellant thus submits that the cited references do not teach the limitations recited in claim 29, and thus, claim 29 is patentable over McKaskle, Lee, and Yamamoto.

Claim 30

Claim 30 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 30 adds to claim 29 the limitations of, "displaying on the screen a hardware refnum node in response to user input, wherein the hardware refnum node references the description of the hardware device; and connecting the hardware refnum node to the first node, wherein said connecting provides the first node with a reference to the description of the hardware device."

The Examiner states that claim 30 is of the same scope as claim 12. Appellant respectfully disagrees and submits that claim 30 recites different limitations than claim 12. Appellant also submits that the cited references do not teach connecting a hardware refnum node to a first node, wherein the hardware refnum node provides the first node with a reference to a description of a hardware device, and wherein the first node is connected to a register access node and provides the description of the hardware device to the register access node.

Appellant thus submits that the cited references do not teach the limitations recited in claim 30, and thus, claim 30 is patentable over McKaskle, Lee, and Yamamoto.

Claim 33

Claim 33 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 33 adds to claim 17 the limitations of, "constructing execution instructions in response to the graphical program, wherein the execution instructions are executable to access hardware registers of the hardware device."

The Examiner states that claim 33 is of the same scope as claim 24. Appellant respectfully disagrees and submits that claim 33 recites different limitations than claim 24. Appellant also submits that none of the cited references teach the additional limitations recited in claim 33, and thus, claim 33 is patentable over McKaskle, Lee, and Yamamoto.

Claim 34

Claim 34 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 34 adds to claim 33 the limitations of, "executing said execution instructions, wherein the register access node accesses hardware registers of the hardware device during said executing."

The Examiner asserts that "it" teaches the limitations recited in claim 34, citing Col. 10,.lines 26-37. It is unclear which reference is referred to as "it". However, none of the cited references teach a register access node accessing hardware registers of a hardware device during execution of a graphical program. Appellant thus submits that claim 34 is patentable over McKaskle, Lee, and Yamamoto.

Claim 35

Claim 35 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 35 adds to claim 17 the limitation of, "wherein the graphical program is operable to access hardware registers of the hardware device for performing instrumentation functions on an instrument."

The Examiner asserts that this limitation of claim 35 is taught by McKaskle at Col. 10, lines 26-37. However, the cited portion of McKaskle refers to the brief descriptions of FIGS. 133-140. This portion of McKaskle teaches nothing whatsoever about a graphical program accessing hardware registers of a hardware device, and more particularly, teaches nothing about accessing hardware registers of a hardware device for performing instrumentation functions on an instrument. McKaskle simply does not teach the specific limitation recited in claim 35. Appellant thus submits that claim 35 is patentable over McKaskle, Lee, and Yamamoto.

Section 103(a) Rejections Based on McKaskle, Lee, Yamamoto, and McIntyre

Claims 19-22 and 26-28 stand rejected under 35 U.S.C. 103(a) as being unpatentable over McKaskle in view of Lee, further in view of Yamamoto, and further in view of McIntyre et al. (U.S. Patent No. 6,229,538, hereinafter "McIntyre"). Appellant respectfully traverses this rejection.

Claim 19

Claim 19 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 19 recites, "The method of claim 18, wherein said configuring the register access node includes: displaying a list of hardware registers described in the description of the hardware device; receiving user input to select one or more of the hardware registers from the list of hardware registers." The Examiner asserts that these limitations are taught by McIntyre in FIG. 5 and Col. 8, lines 38-64. Appellant disagrees.

McIntyre relates generally to computer networking systems, and more particularly to a method and apparatus for providing port-centric graphic representations of network controllers (Col. 1, lines 6-9). FIG. 5 illustrates an embodiment in which an intermediate driver defines a Heartbeat Multicast Address (HMC) and causes each NIC (network interface card) team member to register the HMC address. A network multicast address is not the same as a hardware register of a hardware device. A list of hardware registers of a hardware device is not displayed in FIG. 5. Furthermore, McIntyre teaches nothing about a description of a hardware device, wherein the description describes a list of hardware registers in the hardware device. McIntyre also teaches nothing about receiving user input to select one or more hardware registers from a displayed list of hardware registers.

Appellant thus submits that the cited references do not teach the limitations recited in claim 19, and thus, claim 19 is patentable over McKaskle, Lee, Yamamoto, and McIntyre.

Claim 20 and 22

Claims 20 and 22 are separately patentable over the cited references because the references do not teach or suggest the further limitations recited in these claims. For example, claim 20 adds to claim 19 the limitations of:

> wherein the description includes mnemonic names of hardware registers;
> wherein said configuring the register access node includes:
> displaying a list of the mnemonic names of hardware registers on the display; and
> receiving user input selecting one or more of the mnemonic names of hardware registers for access.

The Examiner asserts that these limitations are taught by McIntyre in FIG. 5 (items D1-D4) and at Col. 8, lines 36-64. However, as discussed above with reference to claim 19, these portions relate to an embodiment of McIntyre's system in which an intermediate driver defines a Heartbeat Multicast Address (HMC) and causes each NIC (network interface card) team member to register the HMC address. The items D1-D4 which the Examiner refers to are network interface card (NIC) drivers, not mnemonic names of hardware registers.

Appellant thus submits that the cited references do not teach the limitations recited in claim 20, and thus, claim 20 is patentable over McKaskle, Lee, Yamamoto, and McIntyre. Appellant also submits that claim 22 is patentable over McKaskle, Lee, Yamamoto, and McIntyre for similar reasons. As noted above, the items D1-D4 which the Examiner refers to are network interface card (NIC) drivers, not mnemonic names of fields in hardware registers.

Claim 21

Claim 21 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 21 adds to claim 20 the limitation of, "displaying selected mnemonic names of hardware registers on the display after said receiving user input selecting one or more of the mnemonic names of hardware registers for access."

The Examiner asserts that this limitation is taught by McKaskle in FIG. 100B and refers to the item labeled "7". However, FIG. 100 simply illustrates the operation of initialized and uninitialized shift registers for a While loop structure or For loop structure (Col. 9, lines 26-27). The item labeled "7" in FIG. 100B is simply a numeric value, not a mnemonic name of a hardware register. McKaskle teaches nothing whatsoever about displaying selected mnemonic names of hardware registers on a display after receiving user input selecting the mnemonic names.

Appellant thus submits that the cited references do not teach the limitations recited in claim 21, and thus, claim 21 is patentable over McKaskle, Lee, Yamamoto, and McIntyre.

Claim 26

Claim 26 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim, for reasons similar to those discussed above with reference to claims 19, 20, and 22.

Claim 27

Claim 27 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 27 adds to claim 26 the limitation of, "wherein the register access node is a growable node which may comprise a variable number of user selected terminals." The Examiner asserts that this limitation is taught by McKaskle in FIG. 100B and refers to the item labeled "7". However, McKaskle contains no teaching whatsoever about this item being a growable node which may comprise a variable number of user selected terminals.

Appellant thus submits that the cited references do not teach the limitation recited in claim 27, and thus, claim 27 is patentable over McKaskle, Lee, Yamamoto, and McIntyre.

Claim 28

Claim 28 is separately patentable over the cited references because the references do not teach or suggest the further limitations recited in this claim. Claim 28 adds to claim 17 the limitation of, "displaying a name of the hardware description on the register access node in response to connecting the first node to the register access node." The Examiner states that claim 28 is of the same scope as claim 26. Appellant respectfully disagrees and submits that claim 28 recites different limitations that claim 26. Appellant also submits that the cited references do not teach the limitation recited in claim 28, for reasons similar to those discussed with reference to various claims above. Appellant thus submits that claim 28 is patentable over McKaskle, Lee, Yamamoto, and McIntyre.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-48 was erroneous, and reversal of the Examiner's decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5150-38200/JCH. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

Jeffrey C. Hood
Reg. No. 35,198
ATTORNEY FOR APPLICANT(S)

Meyertons Hood Kivlin Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: 4/24/2006 JCH/JLB

IX. CLAIMS APPENDIX

The following lists the claims as incorporating entered amendments, and as on appeal.

1. (Previously Presented) A method for creating a graphical program which performs hardware register accesses in a hardware device, wherein the method operates in a computer including a display screen and a user input device, the method comprising:

displaying on the screen a register access node in the graphical program in response to user input; and

configuring the register access node to access one or more hardware registers of the hardware device, wherein said configuring includes accessing a description of the hardware device for information regarding the one or more hardware registers of the hardware device;

wherein, during execution of the graphical program, the register access node is operable to access the one or more hardware registers of the hardware device based on the information.

2. (Previously Presented) The method of claim 1, wherein said configuring the register access node includes:

displaying a list of hardware registers; and

receiving user input to select one or more of the hardware registers from the list of hardware registers.

3. (Previously Presented) The method of claim 1, further comprising:

storing the description of the hardware device in the computer;

wherein the register access node uses the description of the hardware device to access hardware registers of the hardware device during execution of the graphical program.

4. (Previously Presented) The method of claim 3,

wherein said configuring the register access node to access one or more hardware registers of the hardware device comprises configuring the register access node to access selected hardware registers described in the description of the hardware device.

5. (Previously Presented) The method of claim 3, wherein the description includes mnemonic names of hardware registers;

wherein said configuring the register access node includes:

displaying a list of the mnemonic names of hardware registers on the display; and

receiving user input selecting one or more of the mnemonic names of hardware registers for access.

6. (Previously Presented) The method of claim 5, wherein the description further includes mnemonic names of fields in the hardware registers;

wherein said configuring the register access node includes:

displaying a list of the mnemonic names of fields in the hardware registers on the display; and

receiving user input selecting one or more of the mnemonic names of fields in the hardware registers for access.

7. (Original) The method of claim 1, wherein said configuring the register access node comprises:

displaying an icon on the graphical program which references register access node configuration information; and

connecting the icon to the register access node.

8. (Original) The method of claim 7, wherein the icon is a hardware open node.

9. (Previously Presented) The method of claim 1, wherein said register access node comprises one or more input terminals, the method further comprising:

configuring the one or more input terminals to write to a hardware register of the hardware device.

10. (Previously Presented) The method of claim 1, wherein said register access node comprises one or more output terminals, the method further comprising:

configuring the one or more output terminals to read a hardware register of the hardware device.

11. (Original) The method of claim 1, further comprising:

displaying on the screen a first node in response to user input, wherein the first node references the hardware device; and

connecting the first node to the register access node, wherein said connecting provides the register access node with information regarding the hardware device.

12. (Previously Presented) The method of claim 11, wherein the first node is a hardware refnum node which references the description of the hardware device.

13. (Original) The method of claim 11, wherein said connecting the first node to the register access node includes displaying on the screen a wire connecting the first node to the register access node.

14. (Previously Presented) The method of claim 1, further comprising:

constructing execution instructions in response to the graphical program, wherein the execution instructions are executable to access hardware registers of the hardware device.

15. (Previously Presented) The method of claim 14, further comprising:

executing said execution instructions, wherein the register access node accesses hardware registers of the hardware device during said executing.

16. (Previously Presented) The method of claim 1, wherein the graphical program is operable to access hardware registers of the hardware device for performing instrumentation functions on an instrument.

17. (Previously Presented) A method for creating a graphical program which performs hardware register accesses in a hardware device, wherein the method operates in a computer including a display and a user input device, the method comprising:

storing a description of the hardware device;

displaying on the screen a first node in response to user input, wherein the first node references the description of the hardware device;

displaying on the screen a register access node in response to user input, wherein the register access node is operable to access the hardware device;

connecting the first node to the register access node in response to user input, wherein the first node is operable to provide the description of the hardware device to the register access node;

wherein the register access node receives the description, wherein the register access node is operable to access hardware registers of the hardware device during execution of the graphical program based on the description of the hardware device.

18. (Previously Presented) The method of claim 17, further comprising:

configuring the register access node to access selected hardware registers described in the description of the hardware device, wherein said configuring includes accessing the description of the hardware device for information regarding the selected hardware registers.

19. (Previously Presented) The method of claim 18, wherein said configuring the register access node includes:

displaying a list of hardware registers described in the description of the hardware device;

receiving user input to select one or more of the hardware registers from the list of hardware registers.

20. (Previously Presented) The method of claim 19, wherein the description includes mnemonic names of hardware registers;

wherein said configuring the register access node includes:

displaying a list of the mnemonic names of hardware registers on the display; and

receiving user input selecting one or more of the mnemonic names of hardware registers for access.

21. (Previously Presented) The method of claim 20, further comprising:

displaying selected mnemonic names of hardware registers on the display after said receiving user input selecting one or more of the mnemonic names of hardware registers for access.

22. (Previously Presented) The method of claim 20, wherein the description further includes mnemonic names of fields in the hardware registers;

wherein said configuring the register access node includes:

displaying a list of the mnemonic names of fields in the hardware registers on the display; and

receiving user input selecting one or more of the mnemonic names of fields in the hardware registers for access.

23. (Previously Presented) The method of claim 17, further comprising:

displaying on the screen a list of descriptions of available hardware devices;

selecting the description from the list of descriptions.

24. (Previously Presented) The method of claim 17, wherein said register access node comprises one or more input terminals, wherein, for each input terminal, the method further comprises:

configuring each input terminal to write to a hardware register of the hardware device.

25. (Previously Presented) The method of claim 17, wherein said register access node comprises one or more output terminals, wherein, for each output terminal, the method further comprises:

configuring said each output terminal to read a hardware register of the hardware device.

26. (Previously Presented) The method of claim 17, wherein receiving user input further comprises:

selecting a first hardware register from said list of hardware registers;

associating a first terminal of the register access node with said first hardware register;

selecting the first terminal as a read or a write terminal;

connecting the first terminal to a node in the graphical program; and

repeating the above steps for one or more hardware registers of the hardware device.

27. (Original) The method of claim 26, wherein the register access node is a growable node which may comprise a variable number of user selected terminals.

28. (Original) The method of claim 17, further comprising:

displaying a name of the hardware description on the register access node in response to connecting the first node to the register access node.

29. (Original) The method of claim 17, wherein the first node is a hardware open node.

30. (Original) The method of claim 29, further comprising:

displaying on the screen a hardware refnum node in response to user input, wherein the hardware refnum node references the description of the hardware device; and connecting the hardware refnum node to the first node, wherein said connecting provides the first node with a reference to the description of the hardware device.

31. (Original) The method of claim 17, wherein the first node is a hardware refnum node which references the description of the hardware device.

32. (Original) The method of claim 17, wherein said connecting the first node to the register access node includes displaying on the screen a wire connecting the first node to the register access node.

33. (Previously Presented) The method of claim 17, further comprising:

constructing execution instructions in response to the graphical program, wherein the execution instructions are executable to access hardware registers of the hardware device.

34. (Original) The method of claim 33, further comprising:

executing said execution instructions, wherein the register access node accesses hardware registers of the hardware device during said executing.

35. (Previously Presented) The method of claim 17, wherein the graphical program is operable to access hardware registers of the hardware device for performing instrumentation functions on an instrument.

36. (Previously Presented) A method for creating a graphical program which performs hardware register accesses in a hardware device, wherein the method operates in a computer including a display and a user input device, the method comprising:

storing a description of the hardware device;

displaying on the screen a register access node in the graphical program in response to user input, wherein the register access node is operable to access the hardware device;

connecting an input of the register access node to receive the description of the hardware device in response to user input; and

configuring the register access node to access selected, hardware registers described in the description of the hardware device in response to user input, wherein said configuring includes accessing the description of the hardware device for information regarding the selected hardware registers of the hardware device;

wherein the register access node is operable to access the selected hardware registers of the hardware device during execution of the graphical program based on the information.

37. (Previously Presented) A memory medium for performing hardware register accesses in a hardware device, the memory medium comprising program instructions executable by a processor to:

display on the screen a register access node in the graphical program in response to user input; and

configure the register access node to access one or more hardware registers of the hardware device, wherein, in configuring the register access node to access the one or more hardware registers of the hardware device, the program instructions are executable by the processor to access a description of the hardware device for information regarding the one or more hardware registers of the hardware device;

wherein, during execution of the graphical program, the register access node is operable to access the one or more hardware registers of the hardware device based on the information.

38. (Previously Presented) The memory medium of claim 37, wherein in performing said configuring the register access node, the program instructions are executable by the processor to:

display a list of hardware registers; and

receive user input to select one or more of the hardware registers from the list of hardware registers.

39. (Previously Presented) The memory medium of claim 37, further comprising program instructions executable by the processor to:

store the description of the hardware device in the computer;

wherein the register access node uses the description of the hardware device to access hardware registers of the hardware device during execution of the graphical program.

40. (Previously Presented) The memory medium of claim 39,

wherein said configuring the register access node to access one or more hardware registers of the hardware device comprises configuring the register access node to access selected hardware registers described in the description of the hardware device.

41. (Original) The memory medium of claim 37, wherein said configuring the register access node comprises:

displaying an icon on the graphical program which references register access node configuration information; and

connecting the icon to the register access node in response to user input.

42. (Previously Presented) The memory medium of claim 37, further comprising program instructions executable by the processor to:

construct execution instructions in response to the graphical program, wherein the execution instructions are executable to access hardware registers of the hardware device.

43. (Previously Presented) A system for performing hardware register accesses in a hardware device, the system comprising:

a computer including a processor coupled to a memory; and

a hardware device coupled to the computer;

wherein the processor is operable to execute program instructions stored in the memory to:

display on the screen a register access node in a graphical program in response to user input; and configure the register access node to access one or more hardware registers of the hardware device, wherein, in configuring the register access node to access the one or more hardware registers of the hardware device, the processor is operable to execute the program instructions to access a description of the hardware device for information regarding the one or more hardware registers of the hardware device;

wherein, during execution of the graphical program, the register access node is operable to access the one or more hardware registers of the hardware device based on the information.

44. (Previously Presented) The system of claim 43, wherein in performing said configuring the register access node, the processor is operable to execute program instructions to:

display a list of hardware registers; and

receive user input to select one or more of the hardware registers from the list of hardware registers.

45. (Previously Presented) The system of claim 43, wherein the processor is further operable to execute program instructions stored in the memory to:

store the description of the hardware device;

wherein the register access node uses the description of the hardware device to access hardware registers of the hardware device during execution of the graphical program.

46. (Previously Presented) The system of claim 45,

wherein said configuring the register access node to access one or more hardware registers of the hardware device comprises configuring the register access node to access selected hardware registers described in the description of the hardware device.

47. (Previously Presented) The system of claim 43, wherein the processor is further operable to execute program instructions stored in the memory to:

construct execution instructions in response to the graphical program, wherein the execution instructions are executable to access hardware registers of the hardware device.

48. (Previously Presented) A method for creating a graphical program which performs hardware register accesses in a hardware device, wherein the method operates in a computer including a display screen and a user input device, the method comprising:

displaying on the screen a register access node in the graphical program in response to user input; and

configuring the register access node to access one or more hardware registers of the hardware device;

wherein, during execution of the graphical program, the register access node is operable to access the one or more hardware registers of the hardware device.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.